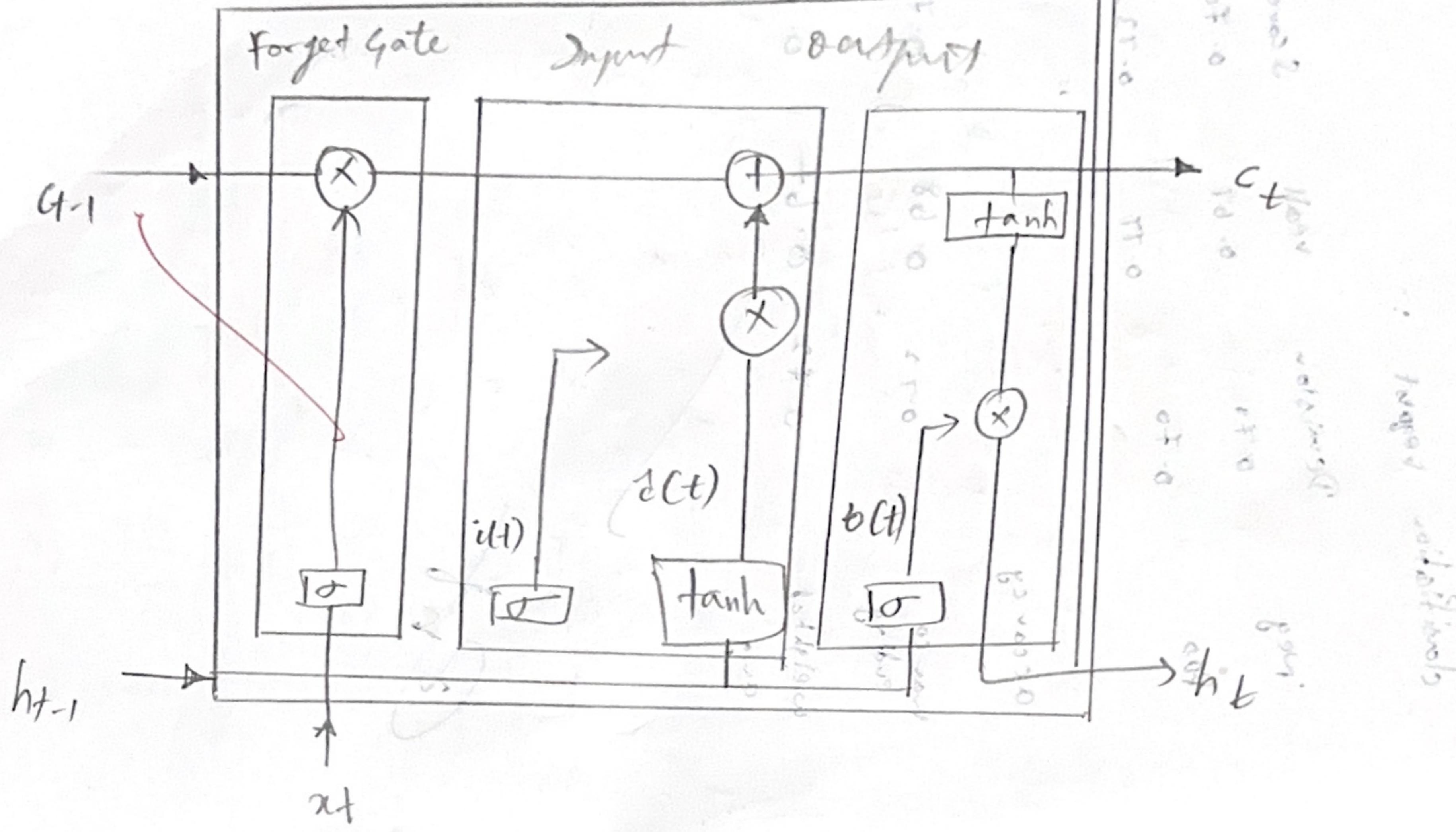


## Objectives

### LSTM Long Short Term Memory

### LSTM Architecture with Coding



[Cell States]  $\rightarrow$

[Cell States]  $\rightarrow$

Input symbols

forget gate  
Input  
hidden state

Aim :-

To implement and train long short term memory network using Pytorch by classifying the number sequences as either increasing or decreasing.

Objective :-

To understand how LSTM process sequential data.

To build and train an LSTM model for classification tasks.

To evaluate model accuracy and visualize predictions.

To analyze the performance w.r.t.

~~LSTM - oriented pattern recognition w.r.t tasks.~~

~~Video 1: Introduction to LSTM~~  
~~Video 2: LSTM Model~~  
~~Video 3: LSTM Model Implementation~~  
~~Video 4: LSTM Model Implementation~~  
~~Video 5: LSTM Model Implementation~~

# PyTorch Text Classification

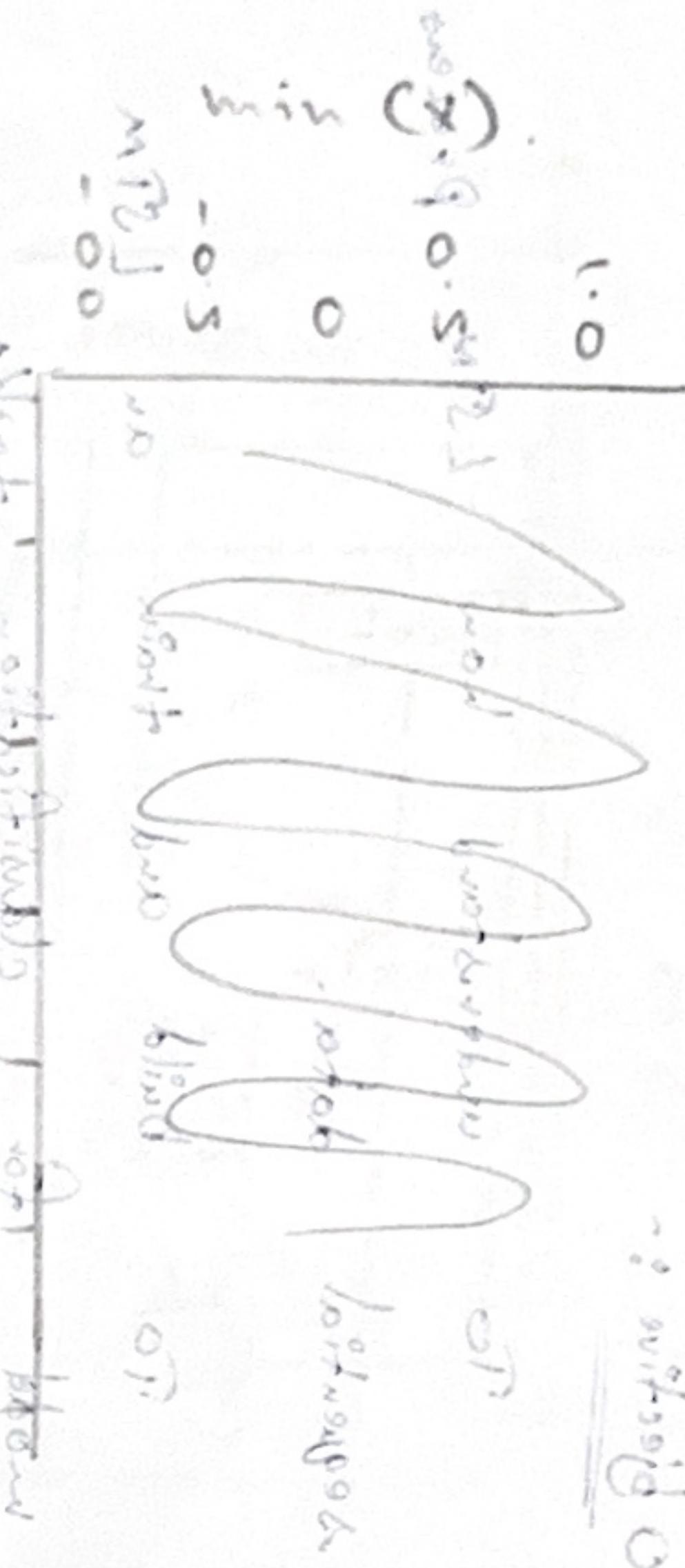
## Procedure:-

- Import necessary libraries.
- Generates Synthetic dataset.
- Pre process data.
- Define LSTM model. (Input-size, hidden size).
- Define loss function and optimizer.
- Cross entropy loss.
- Adam optimizer.
- Train the model.

ratio: 0.85  
 Train: 0.542  
 Validation: 0.542  
 Test: 0.542

Min -0.5  
 Max 0.5

Epochs: 100



Input shape : torch.Size([1960, 30, 1])  
 Target shape : torch.Size([1960, 10, 1]).

Training Loss: 0.05100 OPT.

Epoch [1/30], loss: 0.04980  
 Epoch [2/30], loss: 0.0462.  
 Epoch [3/30], loss: 0.0355.  
 Epoch [4/30], loss: 0.0291.  
 Epoch [5/30], loss: 0.0247.  
 Epoch [6/30], loss: 0.0212.

Observation:-  
 (i) For Paws.  
 (ii) Compute loss.  
 (iii) Backward pass.  
 (iv) Print loss and accuracy every few epochs.

Epoch Training loss Training Accuracy (%)

0	0.6201	42.3
20	0.4052	80.1
40	0.2103	90.7
60	0.0982	95.5
80	0.0524	97.8
100	0.0310	99.0

